LASF – Automates finis déterministes

Christophe Mouilleron

Plan

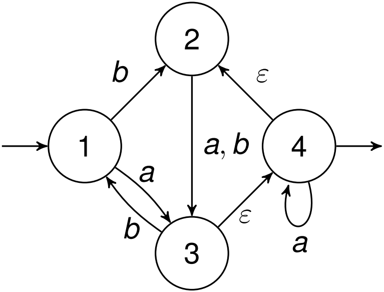
Plan

Transitions vides

Jusqu'à présent : transitions étiquetés par des lettres

Idée : Autoriser l'étiquetage par ε = transition vide.

Exemple :



Avantages et inconvénients des transitions vides

Avantages :

- facilite la conception d'un automate à partir d'un langage ou d'une expression rationnelle
-

Inconvénients :

-
-
-
-

Avantages et inconvénients des transitions vides

Avantages :

- facilite la conception d'un automate à partir d'un langage ou d'une expression rationnelle
- en particulier, rend les opérations produit et étoile plus simples

Inconvénients :

-
-
-
-

Avantages et inconvénients des transitions vides

Avantages :

- facilite la conception d'un automate à partir d'un langage ou d'une expression rationnelle
- en particulier, rend les opérations produit et étoile plus simples

Inconvénients :

- reconnaissance plus complexe à vérifier à la main et difficile à programmer sur machine
-
-
-

Avantages et inconvénients des transitions vides

Avantages :

- facilite la conception d'un automate à partir d'un langage ou d'une expression rationnelle
- en particulier, rend les opérations produit et étoile plus simples

Inconvénients :

- reconnaissance plus complexe à vérifier à la main et difficile à programmer sur machine
- ε n'est **PAS** une lettre, ce qui rend les preuves complexes
-
-

Avantages et inconvénients des transitions vides

Avantages :

- facilite la conception d'un automate à partir d'un langage ou d'une expression rationnelle
- en particulier, rend les opérations produit et étoile plus simples

Inconvénients :

- reconnaissance plus complexe à vérifier à la main et difficile à programmer sur machine
- ε n'est **PAS** une lettre, ce qui rend les preuves complexes
- source de non-déterminisme
- n'apporte pas d'expressivité supplémentaire

Algorithme de suppression des transitions vides

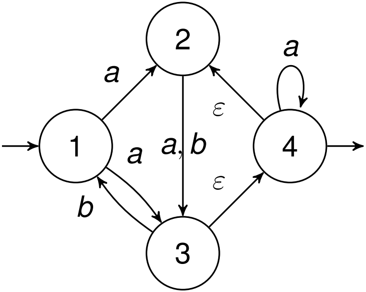
Théorème

Tout automate fini avec transitions vides est équivalent à un automate fini avec autant d'états et sans transitions vides.

Suppression des transitions vides :

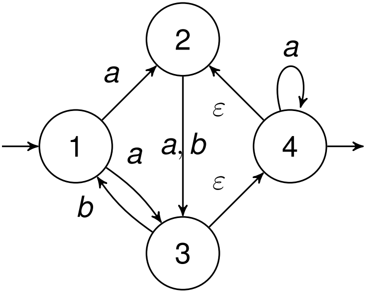
1. calculer les clôtures par transition vide :
$$\mathrm{cl}_\varepsilon(q) = \bigcup_{n \geq 0} \{p \in Q,\ q \xrightarrow{\varepsilon} q_1 \xrightarrow{\varepsilon} \dots \xrightarrow{\varepsilon} q_n = p\}$$
2. pour chaque état q tel que $\mathrm{cl}_\varepsilon(q) \neq \{q\}$:
   - pour tout $(p, a, p') \in E$ avec $p \in \mathrm{cl}_\varepsilon(q)$, $p \neq q$, $a \neq \varepsilon$, ajouter la transition (q, a, p')
   - si $\mathrm{cl}_\varepsilon(q) \cap F \neq \emptyset$, ajouter q à F
3. supprimer toutes les transitions vides

Suppression des transitions vides – Exemple



1. calcul des clôtures par transition vide

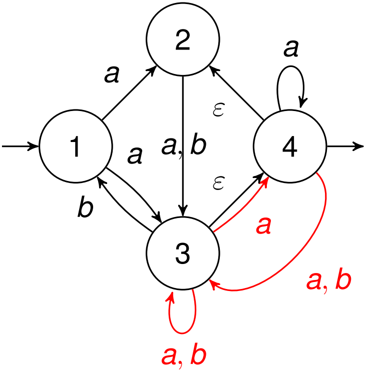
Suppression des transitions vides – Exemple



1. calcul des clôtures par transition vide

$$\mathrm{cl}_\varepsilon(1) = \{1\} \qquad \mathrm{cl}_\varepsilon(2) = \{2\}$$
$$\mathrm{cl}_\varepsilon(3) = \{2, 3, 4\} \qquad \mathrm{cl}_\varepsilon(4) = \{2, 4\}$$

Suppression des transitions vides – Exemple



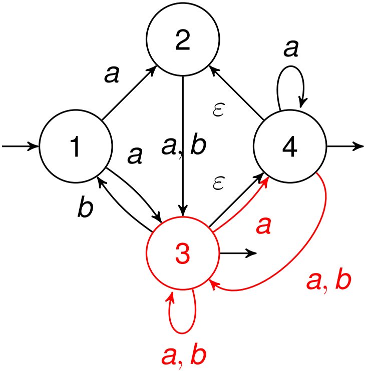
1. calcul des clôtures par transition vide

$$\mathrm{cl}_\varepsilon(1) = \{1\} \qquad \mathrm{cl}_\varepsilon(2) = \{2\}$$
$$\mathrm{cl}_\varepsilon(3) = \{2, 3, 4\} \qquad \mathrm{cl}_\varepsilon(4) = \{2, 4\}$$

2. ajout de transitions

ex : $\left.\begin{array}{l} 2 \in \mathrm{cl}_\varepsilon(4) \\ (2, a, 3) \in E \end{array}\right\}$ ajout de $(4, a, 3)$

Suppression des transitions vides – Exemple



1. calcul des clôtures par transition vide

$$\mathrm{cl}_\varepsilon(1) = \{1\} \qquad \mathrm{cl}_\varepsilon(2) = \{2\}$$
$$\mathrm{cl}_\varepsilon(3) = \{2, 3, 4\} \qquad \mathrm{cl}_\varepsilon(4) = \{2, 4\}$$

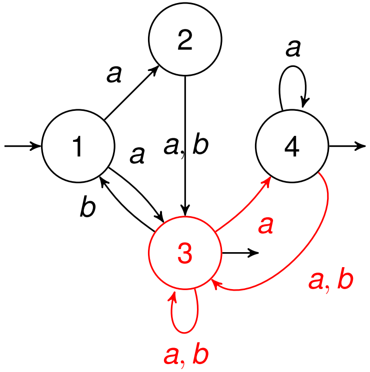
2. ajout de transitions

ex : $\left.\begin{array}{l} 2 \in \mathrm{cl}_\varepsilon(4) \\ (2, a, 3) \in E \end{array}\right\}$ ajout de $(4, a, 3)$

mise à jour des états finals

$$4 \in \mathrm{cl}_\varepsilon(3) \cap F \Rightarrow 3 \text{ devient final}$$

Suppression des transitions vides – Exemple



1. calcul des clôtures par transition vide

$$\mathrm{cl}_\varepsilon(1) = \{1\} \qquad \mathrm{cl}_\varepsilon(2) = \{2\}$$
$$\mathrm{cl}_\varepsilon(3) = \{2, 3, 4\} \qquad \mathrm{cl}_\varepsilon(4) = \{2, 4\}$$

2. ajout de transitions

ex : $\left.\begin{array}{l} 2 \in \mathrm{cl}_\varepsilon(4) \\ (2, a, 3) \in E \end{array}\right\}$ ajout de $(4, a, 3)$

mise à jour des états finals

$$4 \in \mathrm{cl}_\varepsilon(3) \cap F \Rightarrow 3 \text{ devient final}$$

3. suppression des transitions vides

Plan

Automate fini déterministe (AFD)

Ce qui change par rapport à un AF quelconque :

-
-
-

Automate fini déterministe (AFD)

Ce qui change par rapport à un AF quelconque :

- pas de transition vide possible
- un seul état initial q_0
- au plus une transition possible par état de départ + lettre

Définition

Un automate fini déterministe est un quintuplet (A, Q, q_0, F, δ) où :

- A est un alphabet, et Q un ensemble d'état,
- $q_0 \in Q$ est l'état initial de l'automate,
- $F \subset Q$ est l'ensemble des états finaux de l'automate,
- $\delta : Q \times A \to Q$ est la fonction (partielle) de transition.

Chemin et reconnaissance dans un AFD

Extension de δ aux mots : par induction

$$\begin{array}{rcl} \delta(q, \varepsilon) & = & q \\ \delta(q, u) & = & \delta(\delta(q, u_1), u_2 u_3 \dots u_n) \end{array}$$

$\rightsquigarrow$ fonction **partielle** $\delta : Q \times A^* \to Q$.

Propriété

Un mot w est reconnu par l'AFD (A, Q, q_0, F, δ) lorsque $\delta(q_0, w) \in F$.

Propriété des AFDs

Propriétés basiques :

- au plus un chemin étant donné un état de départ et un mot
- représentation aisée à l'aide d'une table

Propriétés liées aux opérations sur les automates :

- l'émondé d'un AFD est un AFD
- le complété d'un AFD est un AFD
  - $\rightsquigarrow$ automate fini déterministe complet (AFDC)

Plan

Déterminisation d'un automate fini

Théorème

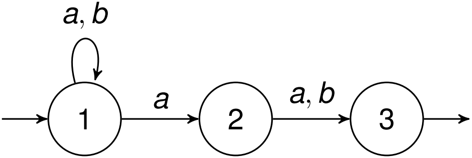
Tout automate fini est équivalent à un automate fini déterministe.

Procédé de déterminisation = transforme un AF $\mathcal{A} = (A, Q, I, F, E)$ en un AFD $\mathcal{A}_D = (A, Q_D, q_0, F, \delta_D)$ équivalent.

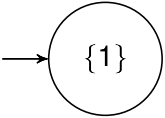
Idée :

- supprimer les transitions vides si besoin,
- considérer l'ensemble des chemins possibles dans $\mathcal{A}$,
- état dans $\mathcal{A}_D$ = sous-ensemble de Q,
- transition dans $\mathcal{A}_D$ = ensemble des transitions possibles dans $\mathcal{A}$.

Exemple :

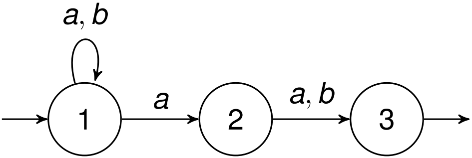


	a	b
1	1, 2	1
2	3	3
3	$\emptyset$	$\emptyset$

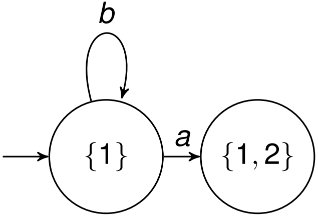


Procédé de déterminisation

Exemple :

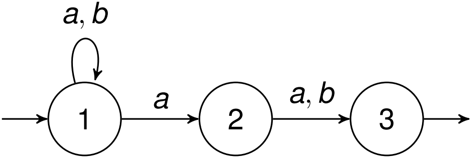


	a	b
1	1, 2	1
2	3	3
3	$\emptyset$	$\emptyset$

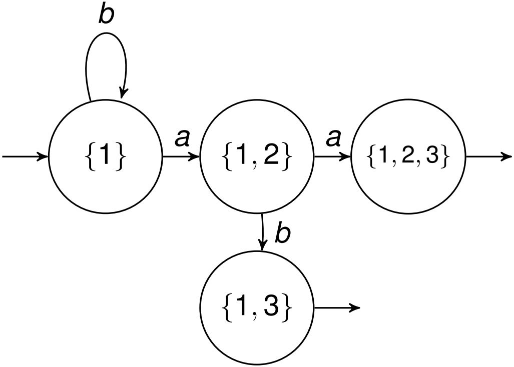


Procédé de déterminisation

Exemple :

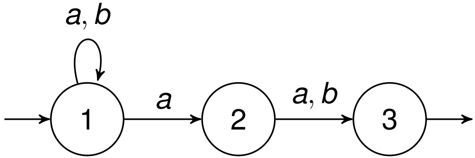


	a	b
1	1, 2	1
2	3	3
3	$\emptyset$	$\emptyset$
1, 2	1, 2, 3	1, 3

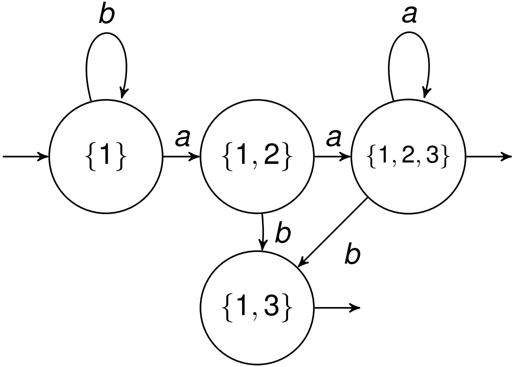


Procédé de déterminisation

Exemple :

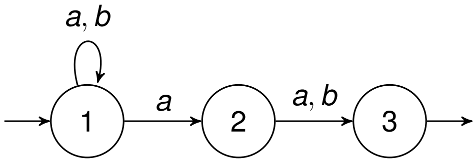


	a	b
1	1, 2	1
2	3	3
3	$\emptyset$	$\emptyset$
1, 2	1, 2, 3	1, 3
1, 2, 3	1, 2, 3	1, 3

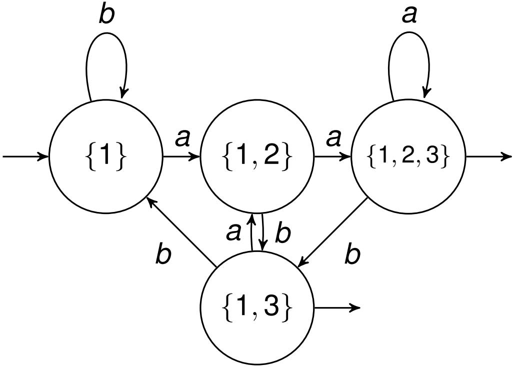


Procédé de déterminisation

Exemple :



	a	b
1	$1, 2$	1
2	3	3
3	$\emptyset$	$\emptyset$
$1, 2$	$1, 2, 3$	$1, 3$
$1, 2, 3$	$1, 2, 3$	$1, 3$
$1, 3$	$1, 2$	1



Application : reconnaissance du complémentaire

Théorème

Si un langage L est reconnaissable par un automate fini, alors le langage L^C l'est aussi.

Application : reconnaissance du complémentaire

Théorème

Si un langage L est reconnaissable par un automate fini, alors le langage L^C l'est aussi.

Reconnaissance du complémentaire :

1. on se donne $\mathcal{A} = (A, Q, I, F, E)$ tel que $\mathrm{L}(\mathcal{A}) = L$,
2. on construit $\mathcal{A}_D = (A, Q_D, q_0, F_D, \delta_D)$ déterministe complet équivalent à A,
3. l'automate $\mathcal{A}_D^C = (A, Q_D, q_0, Q_D \setminus F_D, \delta_D)$ reconnaît L^C.

Complémentaire – Exemple

$L = \{u \in A^*,\ u \text{ ne commence pas par } aa\}$ avec $A = \{a, b\}$.

Construction d'un automate à partir de celui de L^C :

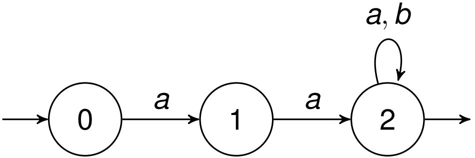
1. automate pour L^C ?

Complémentaire – Exemple

$L = \{u \in A^*,\ u \text{ ne commence pas par } aa\}$ avec $A = \{a, b\}$.

Construction d'un automate à partir de celui de L^C :

1. automate pour $L^C = \{aau,\ u \in A^*\}$:



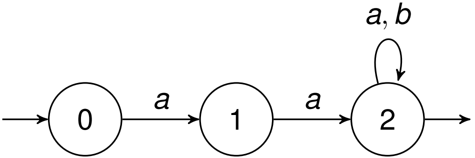
2. passage à un AFDC ici, en déterminisant ou en complétant

Complémentaire – Exemple

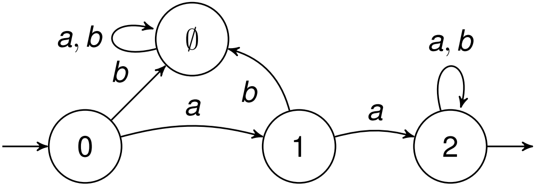
$L = \{u \in A^*,\ u$ ne commence pas par $aa\}$ avec $A = \{a, b\}$.

Construction d'un automate à partir de celui de L^C :

1. automate pour $L^C = \{aau,\ u \in A^*\}$:



2. passage à un AFDC ici, en déterminisant ou en complétant



3. échange des statuts de sortie